Title: SYSTEMS AND METHODS FOR AGGREGATING INFORMATION-ASSET METADATA FROM MULTIPLE DISPARATE DATA-MANAGEMENT SYSTEMS

Abstract: A computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems which may include (1) receiving a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain, (2) receiving a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain, (3) storing the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems, and (4) providing access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity.
Published:
— with international search report (Art. 21(3))
— with amended claims (Art. 19(1))
SYSTEMS AND METHODS FOR AGGREGATING INFORMATION-ASSET METADATA FROM MULTIPLE DISPARATE DATA-MANAGEMENT SYSTEMS

CROSS REFERENCE TO RELATED APPLICATION

This application claims priority to U.S. Provisional Application No. 61/988,213, titled “SYSTEMS AND METHODS FOR AGGREGATING DISPARATE INFORMATION ASSETS SOURCED BY MULTIPLE COLLECTORS AND DATA MANAGEMENT SOLUTIONS” and filed 4 May 2014, the disclosure of which is incorporated, in its entirety, by this reference.

BACKGROUND

In the digital age, organizations and other entities may manage increasingly large volumes of information assets (e.g. files, emails, etc.) across many sources. Some organizations may deploy extensive data-management infrastructures for storing, organizing, protecting, and accessing their information assets. A typical organization’s data-management infrastructure may include multiple disparate data-management systems that manage information assets from a variety of sources. For example, an organization may deploy a backup system that backs up the organization’s information assets, an archiving system that archives the organization’s information assets, and a data-loss-protection system that protects the organization’s information assets from data loss.

Unfortunately, using disparate data-management systems to manage large amounts of information assets across many sources may present unwanted limitations. For example, disparate data-management systems may have different views of the information assets that they manage, may manage information assets in different ways, and/or may manage different sets of information assets. As a result, an organization may be required to access several data-management systems in order to gain an understanding of the organization’s information assets and how they should be managed. Moreover, an organization may be unable to determine if the information assets managed by one data-management system are the same as or related to those managed by another data-management system. Additionally, disparate data-management systems may duplicate effort by performing similar scanning or processing operations on an organization’s information assets. Accordingly, the instant disclosure identifies and addresses a need for additional and improved systems and methods for
aggregating and utilizing information-asset metadata from multiple disparate data-management systems.

SUMMARY

As will be described in greater detail below, the instant disclosure describes various systems and methods for aggregating and utilizing information-asset metadata from multiple disparate data-management systems. In one example, a computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems may include (1) receiving a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain (e.g., a backup domain), (2) receiving a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain (e.g., a data-loss-prevention domain) that is separate and distinct from the first domain, (3) storing the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems such that the first and second instances of metadata are associated with the information asset, and (4) providing access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity. In some embodiments, the first data-management system may have generated (e.g., created or identified) the first instance of metadata, and the second data-management system may have generated the second instance of metadata. In at least one embodiment, the information asset may include a file, an email, and/or a document.

In some embodiments, the first data-management system may manage a first instance of the information asset and the second data-management system may manage a second instance of the information asset such that the first instance of metadata includes metadata of the first instance of the information asset and the second instance of metadata includes metadata of the second instance of the information asset. In one embodiment, the step of storing the first and second instances of metadata in the global metadata repository may include (1) determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset and (2) associating the first and second instances of metadata based on determining that the first
instance of the information asset and the second instance of the information asset are instances of the same information asset.

In another embodiment, the step of storing the first and second instances of metadata in the global metadata repository may include (1) generating a first fingerprint based on the first instance of the information asset and/or the first instance of metadata, (2) generating a second fingerprint based on the second instance of the information asset and/or the second instance of metadata, (3) determining that the first fingerprint matches the second fingerprint, and (4) associating, the first and second instances of metadata based on determining that the first fingerprint matches the second fingerprint.

In some embodiments, the global metadata repository may use a normalized schema to store the first and second instances of metadata. In at least one embodiment, the global metadata repository may use an extensible schema to store the first and second instances of metadata. In at least one embodiment, the extensible schema may enable the first and second data-management systems to store any metadata in the global metadata repository.

In some embodiments, the information asset may be accessible via an information-asset source, the first and second data-management systems may both be capable of generating the first instance of metadata, and the first and second data-management systems may be required to access the information-asset source to generate the first instance of metadata. In at least one embodiment, the method may include coordinating the first and second data-management systems so that (1) the first data-management system accesses the information-asset source, generates the first instance of metadata, and stores the first instance of metadata in the global metadata repository and (2) the second data-management system accesses the first instance of metadata from the global metadata repository without accessing the information-asset source.

In some embodiments, the information asset may be one of a plurality of information assets of the entity that are accessible via an information-asset source, and the steps of receiving the first instance of metadata and receiving the second instance of metadata may include ensuring that metadata of each of the plurality of information assets is received from the first data-management system or the second data-management system. In at least one embodiment, the first data-management system may include a backup system that has generated a backup of the information-asset source, and the step of receiving the first
instance of metadata may include requesting, from the backup system, an inventory of each information asset that is contained in the backup of the information-asset source.

In one embodiment, a system for implementing the above-described method may include (1) a receiving module, stored in memory, that receives (a) a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain and (b) a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain, (2) a storing module, stored in memory, that stores the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems such that the first and second instances of metadata are associated with the information asset, (3) a providing module, stored in memory, that provides access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity, and (4) at least one physical processor configured to execute the receiving module, the storing module, and the providing module.

In some examples, the above-described method may be encoded as computer-readable instructions on a non-transitory computer-readable medium. For example, a computer-readable medium may include one or more computer-executable instructions that, when executed by at least one processor of a computing device, may cause the computing device to (1) receive a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain, (2) receive a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain, (3) store the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems such that the first and second instances of metadata are associated with the information asset, and (4) provide access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity.

In another example, a computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems may include (1)
identifying, at a first data-management system that manages information assets of an entity in a first domain, an information asset, (2) generating, at the first data-management system, a first instance of metadata of the information asset, and (3) storing the first instance of metadata in a global metadata repository that (a) stores a second instance of metadata of the information asset that was collected from a second data-management system and (b) is separate and distinct from the first data-management system. In some embodiments, the second data-management system may have generated the second instance of metadata and may manage the information assets of the entity in a second domain that is separate and distinct from the first domain.

In some embodiments, the information asset may be accessible via an information-asset source, the first data-management system may include a backup system that backs up the information-asset source, and the backup system may generate the first instance of metadata as part of backing up the information-asset source.

In some embodiments, the method may include (1) detecting that the information asset has been modified, (2) updating the first instance of metadata in response to detecting that the information asset has been modified, and (3) storing the updated first instance of metadata to the global metadata repository.

In some embodiments, the method may include (1) requesting, at the first data-management system, information associated with the information asset from the global metadata repository, (2) receiving, at the first data-management system and from the global metadata repository, information associated with the information asset that may include the second instance of metadata, and (3) performing, at the first data-management system, an action involving the information asset based on the information associated with the information asset received from the global metadata repository.

In some embodiments, a system for implementing the above-described method may include (1) an identifying module, stored in memory, that identifies an information asset at a first data-management system that manages information assets of an entity in a first domain, (2) a generating module, stored in memory, that generates, at the first data-management system, a first instance of metadata of the information asset, (3) a storing module, stored in memory, that stores the first instance of metadata in a global metadata repository that (a) stores a second instance of metadata of the information asset that was collected from a second data-management system and (b) is separate and distinct from the first data-
management system and the second data-management system, and (4) at least one physical processor configured to execute the identifying module, the generating module, and the storing module.

In some examples, the above-described method may be encoded as computer-readable instructions on a non-transitory computer-readable medium. For example, a computer-readable medium may include one or more computer-executable instructions that, when executed by at least one processor of a computing device, may cause the computing device to (1) identify an information asset at a first data-management system that manages information assets of an entity in a first domain, (2) generate, at the first data-management system, a first instance of metadata of the information asset, and (3) store the first instance of metadata in a global metadata repository that (a) stores a second instance of metadata of the information asset that was collected from a second data-management system and (b) is separate and distinct from the first data-management system and the second data-management system.

Features from any of the above-mentioned embodiments may be used in combination with one another in accordance with the general principles described herein. These and other embodiments, features, and advantages will be more fully understood upon reading the following detailed description in conjunction with the accompanying drawings and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings illustrate a number of exemplary embodiments and are a part of the specification. Together with the following description, these drawings demonstrate and explain various principles of the instant disclosure.

FIG. 1 is a block diagram of an exemplary system for aggregating information-asset metadata from multiple disparate data-management systems.

FIG. 2 is a block diagram of an additional exemplary system for aggregating information-asset metadata from multiple disparate data-management systems.

FIG. 3 is a flow diagram of an exemplary method for aggregating information-asset metadata from multiple disparate data-management systems.

FIG. 4 is a block diagram of exemplary data-management systems.

FIG. 5 is a block diagram of an exemplary global metadata repository.

FIG. 6 is a block diagram of an exemplary global metadata repository.
FIG. 7 is a block diagram of an exemplary global metadata repository.

FIG. 8 is a flow diagram of an exemplary method for aggregating information-asset metadata from multiple disparate data-management systems.

FIG. 9 is a block diagram of an exemplary computing system capable of implementing one or more of the embodiments described and/or illustrated herein.

FIG. 10 is a block diagram of an exemplary computing network capable of implementing one or more of the embodiments described and/or illustrated herein.

Throughout the drawings, identical reference characters and descriptions indicate similar, but not necessarily identical, elements. While the exemplary embodiments described herein are susceptible to various modifications and alternative forms, specific embodiments have been shown by way of example in the drawings and will be described in detail herein. However, the exemplary embodiments described herein are not intended to be limited to the particular forms disclosed. Rather, the instant disclosure covers all modifications, equivalents, and alternatives falling within the scope of the appended claims.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

The present disclosure is generally directed to systems and methods for aggregating information-asset metadata from multiple disparate data-management systems. As will be explained in greater detail below, by aggregating information-asset metadata of an entity's information assets from multiple disparate data-management systems into a single global metadata repository, the systems and methods described herein may provide a normalized, unified, and consistent view of the entity's information assets to the data-management systems and the entity. In one example, the systems and methods described herein may aggregate information-asset metadata into a single global metadata repository by enabling disparate data-management systems to contribute their knowledge about the information assets to the global metadata repository.

Moreover, by creating the global metadata repository, the systems and methods described herein may enable the entity and the data-management systems to gain, from a single source, a complete understanding of the organization's information assets and how the information assets are and should be managed. Furthermore, by enabling disparate data-management systems to access and enrich a global metadata repository, the systems and methods described herein may enable the disparate data-management systems to efficiently
share metadata such that (1) at least one data-management system is able to discover an entity's information assets that exist across multiple sources without scanning the sources themselves, (2) valuable knowledge about information assets may be shared amongst several data-management systems (e.g., knowledge indicating that an asset is sensitive or valuable), and (3) duplications of effort may be reduced or eliminated. Embodiments of the instant disclosure may also provide various other advantages and features, as discussed in greater detail below.

The following will provide, with reference to FIGS. 1-2, detailed descriptions of exemplary systems for aggregating information-asset metadata from multiple disparate data-management systems. Detailed descriptions of corresponding computer-implemented methods will also be provided in connection with FIGS. 3-8. In addition, detailed descriptions of an exemplary computing system and network architecture capable of implementing one or more of the embodiments described herein will be provided in connection with FIGS. 9 and 10, respectively.

FIG. 1 is a block diagram of an exemplary system 100 for aggregating information-asset metadata from multiple disparate data-management systems. As illustrated in this figure, exemplary system 100 may include one or more modules 102 for performing one or more tasks. For example, and as will be explained in greater detail below, exemplary system 100 may include a receiving module 104 that may (1) receive a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain and (2) receive a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain. Exemplary system 100 may also include a storing module 106 that may store the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems such that the first and second instances of metadata are associated with the information asset. Exemplary system 100 may also include a providing module 108 that may provide access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity.

In addition, and as will be described in greater detail below, exemplary system 100 may include an identifying module 110 that may identify, at the first data-management
system, an information asset. Exemplary system 100 may additionally include a generating module 112 that may generate, at the first data-management system, the first instance of metadata. Although illustrated as separate elements, one or more of modules 102 in FIG. 1 may represent portions of a single module or application.

In certain embodiments, one or more of modules 102 in FIG. 1 may represent one or more software applications or programs that, when executed by a computing device, may cause the computing device to perform one or more tasks. For example, and as will be described in greater detail below, one or more of modules 102 may represent software modules stored and configured to run on one or more computing devices, such as the devices illustrated in FIG. 2 (e.g., data-management systems 202(1)-(N), server 206, and/or information-asset sources 208 and 210), computing system 910 in FIG. 9, and/or portions of exemplary network architecture 1000 in FIG. 10. One or more of modules 102 in FIG. 1 may also represent all or portions of one or more special-purpose computers configured to perform one or more tasks.

As illustrated in FIG. 1, exemplary system 100 may also include a global metadata repository 120 and one or more information assets, such as information asset 140. In one example, global metadata repository 120 may store a representation of one or more information assets (e.g., information assets 122), metadata of one or more information assets (e.g., metadata 124), fingerprints of one or more information assets (e.g., fingerprints 126), information about one or more data-management systems (e.g., data-management systems 128), and/or information about one or more sources of information assets (e.g., information-asset sources 130).

Global metadata repository 120 generally represents a single logical repository of metadata. In some embodiments, global metadata repository 120 may represent portions of a single database or computing device or a plurality of databases or computing devices. For example, global metadata repository 120 may represent a portion of server 206 in FIG. 2, computing system 910 in FIG. 9, and/or portions of exemplary network architecture 1000 in FIG. 10. Alternatively, global metadata repository 120 in FIG. 1 may represent one or more physically separate devices capable of being accessed by a computing device, such as data-management systems 202(1)-(N), server 206, and/or information-asset sources 208 and 210 in FIG. 2, computing system 910 in FIG. 9, and/or portions of exemplary network architecture 1000 in FIG. 10. In some examples, global metadata repository 120 may represent a noSQL
data base such as CASSANDRA, HADOOP, MONGODB, and/or any other suitable type or form of noSQL data base. As used herein, the term "noSQL database" generally refers to a data base that models data in structures other than tabular relations. NoSQL data bases may include data bases that provide wide column stores, document stores, key value stores, graphical relationships, object relationships, tabular relationships, and/or various other storage technologies.

Exemplary system 100 in FIG. 1 may be implemented in a variety of ways. For example, all or a portion of exemplary system 100 may represent portions of exemplary system 200 in FIG. 2. As shown in FIG. 2, system 200 may include data-management systems 202(1)-(N), a server 206, and information-asset sources 208 and 210 in communication via a network 204. In one example, data-management systems 202(1)-(N) may be programmed with one or more of modules 102 and/or may store all or a portion of the data in global metadata repository 120. Additionally or alternatively, server 206 may be programmed with one or more of modules 102 and/or may store all or a portion of the data in global metadata repository 120.

In one embodiment, one or more of modules 102 from FIG. 1 may, when executed by at least one processor of data-management systems 202(1)-(N) and/or server 206, enable data-management systems 202(1)-(N) and/or server 206 to aggregate information-asset metadata from data-management systems 202(1)-(N). For example, and as will be described in greater detail below, identifying module 110 may, as part of data-management system 202(1) that manages information asset 140 in a first domain, identify information asset 140(1) (e.g., an instance of information asset 140 stored to information-asset source 208). Generating module 112 may then, as part of data-management system 202(1), generate metadata 212 of information asset 140(1). Similarly, identifying module 110 may, as part of data-management system 202(N) that manages information asset 140 in a second domain, identify information asset 140(2) (e.g., another instance of information asset 140 stored to information-asset source 210). Generating module 112 may then, as part of data-management system 202(N), generate metadata 214 of information asset 140(2).

Receiving module 104 may, as part of server 206, receive metadata 212 from data-management system 202(1) and metadata 214 from data-management system 202(N). Storing module 106 may then, as part of server 206, store metadata 212 and 214 in global metadata repository 120 that is separate and distinct from data-management systems 202(1)-(N) such that metadata 212 and 214 are associated with information asset 140 (e.g., as
illustrated in FIGS. 5-7). Providing module 108 may provide access to metadata 212 and 214 to one or more of data-management system 202(1)-(N), an owner of information asset 140, and/or an administrator of system 200.

Data-management systems 202(1)-(N) generally represent any type or form of computing device that is capable of reading computer-executable instructions and/or performing data-management operations. Server 206 generally represents any type or form of computing device that is capable of reading computer-executable instructions and/or managing a global metadata repository. Examples of data-management systems 202(1)-(N) and server 206 include, without limitation, application servers and database servers configured to provide various database services and/or run certain software applications.

Network 204 generally represents any medium or architecture capable of facilitating communication or data transfer. Examples of network 204 include, without limitation, an intranet, a Wide Area Network (WAN), a Local Area Network (LAN), a Personal Area Network (PAN), the Internet, Power Line Communications (PLC), a cellular network (e.g., a Global System for Mobile Communications (GSM) network), exemplary network architecture 1000 in FIG. 10, or the like. Network 204 may facilitate communication or data transfer using wireless or wired connections. In one embodiment, network 204 may facilitate communication between data-management systems 202(1)-(N), server 206, and information-asset sources 208 and 210.

Information-asset sources 208 and 210 generally represent any type or form of computing device capable of reading computer-executable instructions and/or any type or form of storage device or system capable of storing and/or managing data. Examples of information-asset sources 208 and 210 include, without limitation, application servers, file servers, email servers, web servers, computing system 910 in FIG. 9, and/or portions of exemplary network architecture 1000 in FIG. 10.

FIG. 3 is a flow diagram of an exemplary computer-implemented method 300 for aggregating information-asset metadata from multiple disparate data-management systems. The steps shown in FIG. 3 may be performed by any suitable computer-executable code and/or computing system. In some embodiments, the steps shown in FIG. 3 may be performed by one or more of the components of system 100 in FIG. 1, system 200 in FIG. 2, computing system 910 in FIG. 9, and/or portions of exemplary network architecture 1000 in FIG. 10.
As illustrated in FIG. 3, at step 302 one or more of the systems described herein may receive a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain. For example, receiving module 104 may, as part of server 206 in FIG. 2, receive metadata 212 of information asset 140 from data-management system 202(1) that manages information assets stored to information-asset source 208.

Receiving module 104 may receive information-asset metadata from a variety of data-management systems that manage an entity's information assets. As used herein, the term "data-management system" generally refers to any system or device that manages information assets and/or generates metadata of information assets. Examples of data-management systems include, without limitation, systems that protect, organize, and/or store information assets (e.g., file systems, email systems, document systems, storage systems, backup systems, archival systems, replication systems, high-availability systems, data-search systems, data-lifecycle-management systems, and virtualization systems) and systems that control access to information assets (e.g., data-loss-prevention systems, identity-authentication systems, access-control systems, encryption systems, policy-compliance systems, risk-reduction systems, intrusion-prevention systems, unstructured-data-governance systems, and electronic-discovery systems). In some examples, the term "data-management system" may refer to a cloud-computing environment that provides various data-management services via the Internet. As used herein, the term "entity" generally refers to any organization (e.g., a business or government unit), individual, group of related individuals, department within an organization, computing device, collection of computing devices, or computing system.

Data-management systems may manage a variety of information assets in a variety of domains. As used herein, the term "information asset" generally refers to any discrete or aggregated representation of electronic information. In some examples, the term "information asset" may refer to any structured, semi-structured, or unstructured information. Examples of information assets include, without limitation, files, emails, documents, messages, databases, database tables, containers, folders, backups, disks, database servers, mail servers, and mail boxes. An entity's information assets may be stored across a variety of information-asset sources (e.g., personal computing devices, file servers,
application servers, email servers, document repositories, collaboration systems, social
networks, and cloud-based storage services).

As part of managing information assets, data-management systems may generate and
maintain a variety of information-asset metadata. As used herein, the term "metadata"
generally refers to any data that describes the characteristics or attributes of an information
asset. Examples of metadata include, without limitation, names, data types, ages, creation
attributes, access attributes, modification attributes, size attributes, location attributes,
relationship attributes, usage attributes, ownership attributes, reputation attributes,
permission attributes, classifications, tags, fingerprints, and life-cycle attributes.

In some examples, the term "metadata" may refer to characteristics or attributes of
the content of an information asset but not the content itself. For example, metadata of a
text document may include a number of words contained in the text document but not the
words themselves. Similarly, the term "metadata" may refer to characteristics or attributes
of an information asset that have been generated from the content of the information asset
but not the content itself. For example, metadata of a text document may include a
classification that indicates that the text document includes sensitive information but not the
sensitive information itself. In at least one example, the term "metadata" may refer to
characteristics or attributes of an information asset that have been generated without
examining the content of the information asset. For example, metadata of a text document
may include a name, a size, or a data type that has been discovered by a data-management
system without examining the content of the text document.

The term "domain," as used herein, generally refers to any area or field of data
management. Examples of data-management domains include, without limitation, a file
domain, an email domain, a document domain, a storage domain, a backup domain, an
archival domain, a replication domain, a high-availability domain, a data-search domain, a
virtualization domain, a data-loss-prevention domain, an identity-authentication domain, an
access-control domain, an encryption domain, a policy-compliance domain, an intrusion-
prevention domain, an unstructured-data-governance domain, an electronic discovery
domain, and/or a cloud-computing domain.

In some examples, two data-management systems may be considered to manage
information assets in separate and distinct domains if (1) the two data-management systems
generate or utilize different sets of information-asset metadata while performing data-
management operations, (2) the two data-management systems generate or utilize similar sets of information-asset metadata to perform different data-management operations, or (3) the two data-management systems manage different sets of information assets. For example, a backup system and a data-loss-prevention system may be considered to operate in different domains (e.g., a backup domain and a data-loss-prevention domain, respectively). Similarly, an archiving system that archives files and an archiving system that archives emails may be considered to operate in different domains (e.g., a file domain and an email domain, respectively).

FIG. 4 illustrates two disparate data-management systems that manage instances of information asset 140 (e.g., information asset 140(1) and information asset 140(2)) in separate and distinct domains and that have generated different sets of metadata for information asset 140 while performing data-management operations on the instances of information asset 140. As illustrated in FIG. 4, data-management system 202(1) may have generated metadata 212, metadata 402, and metadata 404 of information asset 140(1), and data-management system 202(N) may have generated metadata 214, metadata 406, and metadata 408 of information asset 140(2). As will be explained in greater detail below, the systems described herein may aggregate the metadata maintained by data-management system 202(1) and the metadata maintained by data-management system 202(N) into a single global metadata repository that may provide a normalized, unified, and consistent view of information asset 140 (e.g., as illustrated in FIGS. 5-7).

Returning to FIG. 3, receiving module 104 may receive metadata from data-management systems in a variety of ways. In one example, receiving module 104 may receive metadata from a data-management system as part of enabling the data-management system to push metadata to a global metadata repository. For example, receiving module 104 may receive metadata from data-management systems via an application programming interface that enables the data-management systems to add and/or update metadata to global metadata repository 120. In at least one example, the application programming interface may enable a data-management system to identify (e.g., using a fingerprint or hash) the information asset to which a particular instance of metadata belongs.

In some examples, receiving module 104 may enable data-management systems to store information-asset metadata to a global metadata repository using a representational state transfer (REST) architecture that provides create, read, update, and delete (CRUD)
functionalities. In one example, receiving module 104 may use a RESTful architecture to enable data-management systems to create, read, update, and/or delete records of information assets and metadata within global metadata repository 120.

Additionally or alternatively, receiving module 104 may receive metadata from a data-management system by pulling the metadata from the data-management system. For example, receiving module 104 may receive metadata from a data-management system via an application programming interface of the data-management system that enables receiving module 104 to pull information-asset metadata from the data-management system. In at least one example, an application programming interface of a data-management system may enable receiving module 104 to subscribe to a metadata feed (e.g., a feed of newly generated and/or updated metadata) from the data-management system.

In some examples, receiving module 104 may receive metadata from a data-management system as part of discovering and/or monitoring the information-asset sources, the information assets, and/or the information-asset metadata that are managed by the data-management system. In one example, receiving module 104 may identify one or more data-management systems (e.g., data-management systems 128) that manage an entity's information assets and then query each data-management system for information about the information-asset sources, the information assets, and/or the information-asset metadata that are managed by the data-management system. In some examples, the application programming interface may enable receiving module 104 to request metadata of a particular information asset using an identifier of the information asset (such as, e.g., a fingerprint or hash). In at least one example, receiving module 104 may ensure that the metadata that it receives and that is and will be stored in a global metadata repository is complete and up to date by periodically querying the data-management systems.

In addition to receiving metadata from data-management systems, receiving module 104 may gather metadata of information assets by directly scanning the sources of the information assets. In one example, receiving module 104 may scan an information-asset source to identify the information assets stored therein. In some examples, receiving module 104 may periodically scan some or all of the information-asset sources within an entity's environment to ensure that each information asset of an entity is represented in a global metadata repository and/or to ensure that the metadata stored in the global metadata repository is complete and up to date.
In one example, receiving module 104 may identify and gather metadata of the information assets that are stored in a particular information-asset source without scanning the information-asset source by requesting, from a backup system that generated a backup of the information-asset source, an inventory of each information asset that is contained in a backup of the information-asset source (e.g., the most recent backup of the information-asset source). In at least one example, the backup system may generate the inventory from a backup catalog maintained by the backup system. In some examples, the backup catalog of the backup system may also include information-asset metadata, which may be included in the inventory provided to receiving module 104. In some examples, receiving module 104 may detect changes to the information assets stored in the information-asset source (e.g., new information assets, updates, and deletes) by requesting, from the backup system, an inventory of each information asset that has been created, updated, or deleted between the latest backups of the information-asset source. In at least one example, the backup system may generate the inventory by examining the most recent incremental backup of the information-asset source or by comparing the last two full backups of the information-asset source.

In some situations, several data-management systems may be capable of generating the same metadata of an information asset. For at least this reason, receiving module 104 may coordinate the data-management systems so that (1) only one of the data-management systems generate the metadata and attempts to store the metadata in a global metadata repository and (2) the other data-management systems access the metadata from the global metadata repository. In some examples, receiving module 104 may also coordinate data-management systems so that at least one of the data-management systems gathers metadata for each information asset and/or from each information-asset source within an entity’s environment without any duplication of effort.

At step 304, one or more of the systems described herein may receive a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain. For example, receiving module 104 may, as part of server 206 in FIG. 2, receive metadata 214 of information asset 140(2) from data-management system 202(N) that manages information assets stored to information-asset source 210. Step 304 is similar to step 302, therefore, the discussion of step 302 may also apply to step 304.
At step 306, one or more of the systems described herein may store the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems such that the first and second instances of metadata are associated with the information asset. For example, storing module 106 may, as part of server 206 in FIG. 2, store metadata 212 and 214 to global metadata repository 120 such that metadata 212 and 214 are associated with information asset 140 (e.g., as illustrated in FIGS. 5-7).

As used herein, the term "global metadata repository" generally refers to any single logical repository of information-asset metadata that is separate and distinct from at least two data-management systems that contribute and/or access the information-asset metadata stored in the global metadata repository. In some examples, the term "global metadata repository" may refer to a centralized or single-instance database. In other examples, the term "global metadata repository" may refer to a database that is physically distributed. For example, a global metadata repository may be deployed using a backing database (such as, e.g., CASSANDRA) that has multiple nodes and that can replicate metadata within and across datacenters. A global metadata repository may be considered separate and distinct from two data-management systems if neither of the two data-management systems are necessary for the global metadata repository to function. Additionally or alternatively, a global metadata repository may be considered separate and distinct from two data-management systems if failure of either of the two data-management systems does not cause failure of the global metadata repository.

Storing module 106 may store metadata in a global metadata repository in a variety of ways. In one example, storing module 106 may store metadata of an information asset in a global metadata repository by writing the metadata to the global metadata repository and by associating the metadata with a representation of the information asset. Using FIGS. 4 and 5 as an example, storing module 106 may write metadata 212, 402, 404 (e.g., metadata of information asset 140(1) received from data-management system 202(1) as part of step 302) and metadata 214, 406, 408 (e.g., metadata of information asset 140(2) received from data-management system 202(N) as part of step 304) in global metadata repository 120 and may associate metadata 212, 214, 402, 404, 406, and 408 with a single representation of information asset 140 (in this example, information asset 500).
In some situations, the systems described herein may receive metadata of two instances of the same information asset. For example, receiving module 104 may receive metadata 212, 402, 404 of information asset 140(1) (e.g., a first instance of information asset 140) and metadata 214, 406, 408 of information asset 140(2) (e.g., a second instance of information asset 140). In this situation, storing module 106 may store the metadata of both instances of the information asset in a global metadata repository by writing the metadata of both instances of the information asset in the global metadata repository and by associating the metadata of both instances of the information asset with a single representation of the information asset (e.g., as described above in connection with FIGS. 4 and 5).

Additionally or alternatively, storing module 106 may store the metadata of both instances of the information asset in a global metadata repository by (1) writing the metadata of both instances of the information asset in the global metadata repository, (2) associating the metadata of the first instance of the information asset with a single representation of the first instance of the information asset stored in the global metadata repository, (3) associating the metadata of the second instance of the information asset with a single representation of the second instance of the information asset stored in the global metadata repository, and (4) associating the representations of the first and second instances of the information asset. Using FIGS. 4 and 6 as an example, storing module 106 may (1) write metadata 212, 402, 404 and metadata 214, 406, 408 to global metadata repository 120, (2) associate metadata 212, 402, and 404 with a single representation of information asset 140(1) (in this example, information asset 602), (3) associate metadata 214, 406, and 408 with a single representation of information asset 140(2) (in this example, information asset 604), and (4) associate the representations of information asset 140(1) and 140(2) with a global representation of information asset 140 (in this example, information asset 600).

In some examples, storing module 106 may identify the information asset to which to associate an instance of metadata using an identifier (e.g., a fingerprint or hash) that was generated by a data-management system and received with the metadata. Additionally or alternatively, storing module 106 may identify the information asset to which to associate an instance of metadata by generating an identifier (e.g., a fingerprint or hash) and using the generated identifier to identify the information asset. In some examples, a data-management system or storing module 106 may generate an identifier based on the non-metadata contents of the information asset and/or the metadata associated with the information asset.
In at least one example, storing module 106 may determine that two information assets are instances of the same information asset in order to associate the two information assets and/or to associate their metadata. In at least one example, storing module 106 may determine that two information assets are instances of the same information asset based on how similar the metadata of one instance of the information asset is to that of the other instance.

Storing module 106 may store metadata of information assets to a global metadata repository using any suitable storage schema that provides a normalized, unified, and consistent view of an information asset's metadata aggregated from two or more disparate data-management systems, that enables metadata to be associated with information assets, and/or that maintains the relationships between multiple instances of a single information asset. In some examples, storing module 106 may use a normalized schema to store metadata in a global metadata repository. As used herein, the term "normalized schema" generally refers to any database schema that provides a unified and consistent view of an information asset's metadata and/or any database schema that avoids redundancies and inconsistencies in stored metadata.

Additionally or alternatively, storing module 106 may use an extensible schema to store metadata. In one example, storing module 106 may use an extensible schema that includes, for each information asset stored in a global metadata repository, a core set of metadata (e.g., a set of metadata that is common to all information assets stored in the global metadata repository) and/or one or more sets of specialized metadata (e.g., one or more sets of metadata that are specialized for a particular kind of information asset or domain). Using FIGS. 4 and 7 as an example, storing module 106 may store metadata 212, 404, and 214 to a global metadata repository 120 as part of a set of core metadata 702 that is associated with information asset 700 (e.g., a representation of information asset 140) and may store metadata 402, 406, and 408 to a global metadata repository 120 as part of a set of specialized metadata 704 that is also associated with information asset 700.

In at least one example, by using an extensible schema to store metadata, storing module 106 may enable a data-management system to augment the metadata of an information asset represented in a global metadata repository with any attributes that it deems valuable. In some examples, two or more data-management systems may be integrated by configuring the data-management systems to share their metadata via a core
or specialized set of metadata stored in a global metadata repository. In at least one example, a data-management system may share metadata with an entity by storing its metadata to a global metadata repository through which the entity may consume the metadata.

At step 308, one or more of the systems described herein may provide access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity. For example, providing module 108 may, as part of server 206 in FIG. 2, provide access to metadata 212 and 214 stored in global metadata repository 120 to data-management systems 202(1)-(N) and/or an owner of information asset 140.

Providing module 108 may provide access to metadata stored in a global metadata repository in any suitable manner. In one example, providing module 108 may provide access to the metadata stored in a global metadata repository to data-management systems via an application programming interface that provides query and search capabilities. In some examples, providing module 108 may enable a data-management system to access metadata associated with a particular information asset of interest. In at least one example, providing module 108 may enable a data-management system to identify (e.g., using a fingerprint or hash) an information asset for which metadata is requested.

Additionally or alternatively, providing module 108 may enable a data-management system to search for information assets with a particular set of attributes (e.g., a particular file size, location, classification, and/or owner). In at least one example, providing module 108 may enable a data-management system to search for additional instances of a particular information asset.

In some examples, by enabling data-management systems to query and search the metadata stored in a global metadata repository, providing module 108 may enable data-management systems to efficiently discover what information assets exist in information-asset sources without scanning the information-asset sources, which may reduce load on the information-asset sources and/or significantly reduce the time taken by data-management systems to discover information assets. For at least this reason, data-management systems that use a global metadata repository as a source of metadata may scan an information-asset source to discover what information assets exist in the information-asset source only after querying the global metadata repository and determining that this information is not contained in the global metadata repository.
In some examples, by enabling data-management systems to query and search the metadata stored in a global metadata repository, providing module 108 may enable disparate data-management systems to share enriched or calculated metadata. For example, providing module 108 may enable an electronic discovery system or a backup system to determine whether an information asset has been classified as sensitive by a data-loss-prevention system.

In some examples, by enabling data-management systems to query and search the metadata stored in a global metadata repository, providing module 108 may reduce or eliminate duplication of effort amongst the data-management systems by enabling data-management systems that are capable of generating the same metadata (e.g., a classification) to generate the metadata only once for each information asset. For at least this reason, data-management systems that use a global metadata repository as a source of metadata may generate metadata of an information asset only after querying the global metadata repository and determining that the metadata of the information asset is not stored in the global metadata repository.

In some examples, providing module 108 may provide metadata stored in a global metadata repository to an entity by providing an analysis or summary of the metadata stored in the global metadata repository to the entity. In one example, providing module 108 may provide, via a graphical user interface, an analysis of an entity's information assets to the entity that describes all or a portion of the entity's information assets, the locations of the information assets, how the information assets are being used, the value of the information assets, and/or any other metadata-based information about the entity's information assets.

In some examples, providing module 108 may provide, via a graphical user interface, a search tool to the entity that enables the entity to search for information assets based on the metadata stored in a global metadata repository. In one example, such a search tool may enable the entity to search for information assets with particular attributes (e.g., a particular file size, classification, or owner) and/or additional instances of a particular information asset.

In some examples, providing module 108 may enable a data-management system to subscribe to event notifications that are triggered by providing module 108 when an event of interest occurs to the metadata stored in a global metadata repository. For example, providing module 108 may enable a data-management system to subscribe to an event notification that is triggered when the metadata of a particular information asset is updated.
by another data-management system. In another example, providing module 108 may enable
a data-management system to subscribe to an event notification that is triggered when
metadata indicating a new instance of a particular information asset is stored to the global
metadata repository. In at least one example, providing module 108 may enable a data-
management system to subscribe to an event notification that is triggered when an attribute
of an information asset is set to a particular value. For example, providing module 108 may
enable a data-management system to subscribe to an event notification that is triggered
when an information asset is classified as sensitive. Upon completion of step 308, exemplary
method 300 in FIG. 3 may terminate.

As indicated above, the systems and methods described herein may enable disparate
data-management systems to use a global metadata repository to store and access
information-asset metadata. FIG. 8 is a flow diagram of an exemplary computer-implemented
method 800 for storing information-asset metadata to a global metadata repository. The
steps shown in FIG. 8 may be performed by any suitable computer-executable code and/or
computing system. In some embodiments, the steps shown in FIG. 8 may be performed by
one or more of the components of system 100 in FIG. 1, system 200 in FIG. 2, computing
system 910 in FIG. 9, and/or portions of exemplary network architecture 1000 in FIG. 10.

As illustrated in FIG. 8, at step 802, one or more of the systems described herein may
identify, at a data-management system, an information asset. For example, identifying
module 110 may, as part of data-management system 202(1) in FIG. 2, identify information
asset 140(1).

The systems described herein may perform step 802 in any suitable manner. In
general, identifying module 110 may identify an information asset while performing normal
data-management operations of a data-management system. For example, identifying
module 110 may, as part of a backup system, identify a file while backing up files. In another
example, identifying module 110 may, as part of an email-archiving system, identify an email
while archiving emails. In another example, identifying module 110 may, as part of a data-
loss-prevention system, identify an information asset while scanning information assets for
sensitive information.

In some examples, identifying module 110 may identify an information asset while
scanning the information-asset source on which the information asset is stored. In at least
one example, identifying module 110 may identify an information asset by querying a global
metadata repository for information assets stored to a specified information-asset source. Additionally or alternatively, identifying module 110 may identify an information asset by receiving a notification from an information-asset source that indicates that the information asset has been modified (e.g., created, updated, or deleted).

In at least one example, identifying module 110 may identify an information asset by receiving, from a system that manages a global metadata repository, a request for metadata of the information asset. For example, identifying module 110 may, as part of data-management system 202(1) in FIG. 2, identify an information asset by receiving, from receiving module 104 on server 206, a request for metadata of the information asset.

At step 804, one or more of the systems described herein may generate, at the data-management system, a first instance of metadata of the information asset. For example, generating module 112 may, as part of data-management system 202(1) in FIG. 2, generate metadata 212 of information asset 140(1).

The systems described herein may perform step 804 in any suitable manner. In general, generating module 112 may, as part of a data-management system, generate metadata of an information asset while performing normal data-management operations of the data-management system. For example, generating module 112 may, as part of a backup system, generate metadata of a file that indicates that the file is contained in a backup created by the backup system. In another example, generating module 112 may, as part of an email-archiving system, generate metadata of an email that indicates that all or a portion of the email has been archived to an archival storage system. In another example, generating module 112 may, as part of a data-loss-prevention system, generate a classification of an information asset that indicates that the information asset contains sensitive information.

In some examples, generating module 112 may generate metadata of an information asset by discovering the information asset or an attribute of the information asset. For example, generating module 112 may generate metadata of a file by discovering the size of the file or the location of the file. In at least one example, generating module 112 may discover metadata of an information asset without examining the content of the information asset. For example, generating module 112 may discover a name, a size, or a data type of a text document without examining the words contained in the text document.

Additionally or alternatively, generating module 112 may generate metadata of an information asset by creating an attribute of the information asset. For example, generating
module 112 may generate metadata of an information asset by classifying the information asset. In some examples, generating module 112 may generate metadata of an information asset by creating an association between the information asset and another information asset. For example, generating module 112 may generate metadata of an information asset by creating an association between an information asset and a container of the information asset (e.g., a backup). In some examples, generating module 112 may generate metadata of an information asset by updating metadata of the information asset in response to changes to the information asset (e.g., modification and/or deletions).

In some examples, generating module 112 may not generate metadata of an information asset if the metadata is already accessible via a global metadata repository. In one example, generating module 112 may determine whether metadata is or is not already accessible via a global metadata repository by querying the global metadata repository. In some examples, generating module 112 may determine that metadata is already accessible via a global metadata repository by receiving a notification from the global metadata repository that indicates that another data-management system has been tasked with generating the metadata.

At step 806, one or more of the systems described herein may store the first instance of metadata in a global metadata repository. For example, storing module 106 may, as part of data-management system 202(1) in FIG. 2, store metadata 212 in global metadata repository 120.

The systems described herein may perform step 806 in any suitable manner. For example, storing module 106 may, as part of a data-management system, transmit metadata to a system that manages a global metadata repository to be stored to the global metadata repository. In some examples, storing module 106 may transmit the metadata to the system that manages the global metadata repository via an application programming interface of the global metadata repository that enables data-management systems to store information-asset metadata to the global metadata repository. In some examples, storing module 106 may transmit metadata to the system that manages the global metadata repository in response to a request from the system. In some examples, storing module 106 may transmit metadata to the system that manages the global metadata repository in response to a request from the system to push metadata to the system whenever it is created or updated. In at least one example, storing module 106 may, as part of a data-management system, provide an
application programming interface that enables the system that manages the global metadata repository to pull information-asset metadata from the data-management system. Upon completion of step 806, exemplary method 800 in FIG. 8 may terminate.

In some examples, a data-management system may use a centralized global metadata repository as both a repository and a source of its own metadata. Additionally or alternatively, a data-management system may use a global metadata repository as a source of metadata that has been generated by other data-management systems. In at least one example, a data-management system may use metadata that has been generated by other data-management systems and that can be access via the centralized global metadata repository to perform some or all of its data-management operations. For example, a data-management system may use metadata that has been generated by other data-management systems to discover what information assets exist in an information-asset source without scanning the information-asset source. In another example, a backup system may use classification metadata (e.g., metadata that indicates that an information asset is sensitive) that has been generated by a data-loss-prevention system to identify and backup sensitive information assets. In another example, an electronic discovery system may use ownership metadata that has been generated by a system to identify a set of information assets belonging to a particular custodian of interest.

As explained above, by aggregating information-asset metadata of an entity's information assets from multiple disparate data-management systems into a single global metadata repository, the systems and methods described herein may provide a normalized, unified, and consistent view of the entity's information assets to the data-management systems and the entity. In one example, the systems and methods described herein may aggregate information-asset metadata into a single global metadata repository by enabling disparate data-management systems to contribute their knowledge about the information assets to the global metadata repository.

Moreover, by creating the global metadata repository, the systems and methods described herein may enable the entity and the data-management systems to gain, from a single source, a complete understanding of the organization's information assets and how the information assets are and should be managed. Furthermore, by enabling disparate data-management systems to access and enrich a global metadata repository, the systems and methods described herein may enable the disparate data-management systems to efficiently
share metadata such that (1) at least one data-management system is able to discover an entity's information assets that exist across multiple sources without scanning the sources themselves, (2) valuable knowledge about information assets may be shared amongst several data-management systems (e.g., knowledge indicating that an asset is sensitive or valuable), and (3) duplications of effort may be reduced or eliminated.

FIG. 9 is a block diagram of an exemplary computing system 910 capable of implementing one or more of the embodiments described and/or illustrated herein. For example, all or a portion of computing system 910 may perform and/or be a means for performing, either alone or in combination with other elements, one or more of the steps described herein (such as one or more of the steps illustrated in FIG. 3). All or a portion of computing system 910 may also perform and/or be a means for performing any other steps, methods, or processes described and/or illustrated herein.

Computing system 910 broadly represents any single or multi-processor computing device or system capable of executing computer-readable instructions. Examples of computing system 910 include, without limitation, workstations, laptops, client-side terminals, servers, distributed computing systems, handheld devices, or any other computing system or device. In its most basic configuration, computing system 910 may include at least one processor 914 and a system memory 916.

Processor 914 generally represents any type or form of physical processing unit (e.g., a hardware-implemented central processing unit) capable of processing data or interpreting and executing instructions. In certain embodiments, processor 914 may receive instructions from a software application or module. These instructions may cause processor 914 to perform the functions of one or more of the exemplary embodiments described and/or illustrated herein.

System memory 916 generally represents any type or form of volatile or non-volatile storage device or medium capable of storing data and/or other computer-readable instructions. Examples of system memory 916 include, without limitation, Random Access Memory (RAM), Read Only Memory (ROM), flash memory, or any other suitable memory device. Although not required, in certain embodiments computing system 910 may include both a volatile memory unit (such as, for example, system memory 916) and a non-volatile storage device (such as, for example, primary storage device 932, as described in detail.
below). In one example, one or more of modules 102 from FIG. 1 may be loaded into system memory 916.

In certain embodiments, exemplary computing system 910 may also include one or more components or elements in addition to processor 914 and system memory 916. For example, as illustrated in FIG. 9, computing system 910 may include a memory controller 918, an Input/Output (I/O) controller 920, and a communication interface 922, each of which may be interconnected via a communication infrastructure 912. Communication infrastructure 912 generally represents any type or form of infrastructure capable of facilitating communication between one or more components of a computing device. Examples of communication infrastructure 912 include, without limitation, a communication bus (such as an Industry Standard Architecture (ISA), Peripheral Component Interconnect (PCI), PCI Express (PCle), or similar bus) and a network.

Memory controller 918 generally represents any type or form of device capable of handling memory or data or controlling communication between one or more components of computing system 910. For example, in certain embodiments memory controller 918 may control communication between processor 914, system memory 916, and I/O controller 920 via communication infrastructure 912.

I/O controller 920 generally represents any type or form of module capable of coordinating and/or controlling the input and output functions of a computing device. For example, in certain embodiments I/O controller 920 may control or facilitate transfer of data between one or more elements of computing system 910, such as processor 914, system memory 916, communication interface 922, display adapter 926, input interface 930, and storage interface 934.

Communication interface 922 broadly represents any type or form of communication device or adapter capable of facilitating communication between exemplary computing system 910 and one or more additional devices. For example, in certain embodiments communication interface 922 may facilitate communication between computing system 910 and a private or public network including additional computing systems. Examples of communication interface 922 include, without limitation, a wired network interface (such as a network interface card), a wireless network interface (such as a wireless network interface card), a modem, and any other suitable interface. In at least one embodiment, communication interface 922 may provide a direct connection to a remote server via a direct...
link to a network, such as the Internet. Communication interface 922 may also indirectly provide such a connection through, for example, a local area network (such as an Ethernet network), a personal area network, a telephone or cable network, a cellular telephone connection, a satellite data connection, or any other suitable connection.

In certain embodiments, communication interface 922 may also represent a host adapter configured to facilitate communication between computing system 910 and one or more additional network or storage devices via an external bus or communications channel. Examples of host adapters include, without limitation, Small Computer System Interface (SCSI) host adapters, Universal Serial Bus (USB) host adapters, Institute of Electrical and Electronics Engineers (IEEE) 1394 host adapters, Advanced Technology Attachment (ATA), Parallel ATA (PATA), Serial ATA (SATA), and External SATA (eSATA) host adapters, Fibre Channel interface adapters, Ethernet adapters, or the like. Communication interface 922 may also allow computing system 910 to engage in distributed or remote computing. For example, communication interface 922 may receive instructions from a remote device or send instructions to a remote device for execution.

As illustrated in FIG. 9, computing system 910 may also include at least one display device 924 coupled to communication infrastructure 912 via a display adapter 926. Display device 924 generally represents any type or form of device capable of visually displaying information forwarded by display adapter 926. Similarly, display adapter 926 generally represents any type or form of device configured to forward graphics, text, and other data from communication infrastructure 912 (or from a frame buffer, as known in the art) for display on display device 924.

As illustrated in FIG. 9, exemplary computing system 910 may also include at least one input device 928 coupled to communication infrastructure 912 via an input interface 930. Input device 928 generally represents any type or form of input device capable of providing input, either computer or human generated, to exemplary computing system 910. Examples of input device 928 include, without limitation, a keyboard, a pointing device, a speech recognition device, or any other input device.

As illustrated in FIG. 9, exemplary computing system 910 may also include a primary storage device 932 and a backup storage device 933 coupled to communication infrastructure 912 via a storage interface 934. Storage devices 932 and 933 generally represent any type or form of storage device or medium capable of storing data and/or other computer-readable
instructions. For example, storage devices 932 and 933 may be a magnetic disk drive (e.g., a
so-called hard drive), a solid state drive, a floppy disk drive, a magnetic tape drive, an optical
disk drive, a flash drive, or the like. Storage interface 934 generally represents any type or
form of interface or device for transferring data between storage devices 932 and 933 and
other components of computing system 910. In one example, global metadata repository 120
from FIG. 1 may be stored in primary storage device 932.

In certain embodiments, storage devices 932 and 933 may be configured to read from
and/or write to a removable storage unit configured to store computer software, data, or
other computer-readable information. Examples of suitable removable storage units include,
without limitation, a floppy disk, a magnetic tape, an optical disk, a flash memory device, or
the like. Storage devices 932 and 933 may also include other similar structures or devices for
allowing computer software, data, or other computer-readable instructions to be loaded into
computing system 910. For example, storage devices 932 and 933 may be configured to read
and write software, data, or other computer-readable information. Storage devices 932 and
933 may also be a part of computing system 910 or may be a separate device accessed
through other interface systems.

Many other devices or subsystems may be connected to computing system 910.
Conversely, all of the components and devices illustrated in FIG. 9 need not be present to
practice the embodiments described and/or illustrated herein. The devices and subsystems
referenced above may also be interconnected in different ways from that shown in FIG. 9.
Computing system 910 may also employ any number of software, firmware, and/or hardware
configurations. For example, one or more of the exemplary embodiments disclosed herein
may be encoded as a computer program (also referred to as computer software, software
applications, computer-readable instructions, or computer control logic) on a computer-
readable medium. The term "computer-readable medium," as used herein, generally refers
to any form of device, carrier, or medium capable of storing or carrying computer-readable
instructions. Examples of computer-readable media include, without limitation, transmission-
type media, such as carrier waves, and non-transitory-type media, such as magnetic-storage
media (e.g., hard disk drives, tape drives, and floppy disks), optical-storage media (e.g.,
Compact Disks (CDs), Digital Video Disks (DVDs), and BLU-RAY disks), electronic-storage media
(e.g., solid-state drives and flash media), and other distribution systems.
The computer-readable medium containing the computer program may be loaded into computing system 910. All or a portion of the computer program stored on the computer-readable medium may then be stored in system memory 916 and/or various portions of storage devices 932 and 933. When executed by processor 914, a computer program loaded into computing system 910 may cause processor 914 to perform and/or be a means for performing the functions of one or more of the exemplary embodiments described and/or illustrated herein. Additionally or alternatively, one or more of the exemplary embodiments described and/or illustrated herein may be implemented in firmware and/or hardware. For example, computing system 910 may be configured as an Application Specific Integrated Circuit (ASIC) adapted to implement one or more of the exemplary embodiments disclosed herein.

FIG. 10 is a block diagram of an exemplary network architecture 1000 in which client systems 1010, 1020, and 1030 and servers 1040 and 1045 may be coupled to a network 1050. As detailed above, all or a portion of network architecture 1000 may perform and/or be a means for performing, either alone or in combination with other elements, one or more of the steps disclosed herein (such as one or more of the steps illustrated in FIG. 3). All or a portion of network architecture 1000 may also be used to perform and/or be a means for performing other steps and features set forth in the instant disclosure.

Client systems 1010, 1020, and 1030 generally represent any type or form of computing device or system, such as exemplary computing system 910 in FIG. 9. Similarly, servers 1040 and 1045 generally represent computing devices or systems, such as application servers or database servers, configured to provide various database services and/or run certain software applications. Network 1050 generally represents any telecommunication or computer network including, for example, an intranet, a WAN, a LAN, a PAN, or the Internet.

In one example, client systems 1010, 1020, and/or 1030 and/or servers 1040 and/or 1045 may include all or a portion of system 100 from FIG. 1.

As illustrated in FIG. 10, one or more storage devices 1060(1)-(N) may be directly attached to server 1040. Similarly, one or more storage devices 1070(1)-(N) may be directly attached to server 1045. Storage devices 1060(1)-(N) and storage devices 1070(1)-(N) generally represent any type or form of storage device or medium capable of storing data and/or other computer-readable instructions. In certain embodiments, storage devices 1060(1)-(N) and storage devices 1070(1)-(N) may represent Network-Attached Storage (NAS)
devices configured to communicate with servers 1040 and 1045 using various protocols, such as Network File System (NFS), Server Message Block (SMB), or Common Internet File System (CIFS).

Servers 1040 and 1045 may also be connected to a Storage Area Network (SAN) fabric 1080. SAN fabric 1080 generally represents any type or form of computer network or architecture capable of facilitating communication between a plurality of storage devices. SAN fabric 1080 may facilitate communication between servers 1040 and 1045 and a plurality of storage devices 1090(1)-(N) and/or an intelligent storage array 1095. SAN fabric 1080 may also facilitate, via network 1050 and servers 1040 and 1045, communication between client systems 1010, 1020, and 1030 and storage devices 1090(1)-(N) and/or intelligent storage array 1095 in such a manner that devices 1090(1)-(N) and array 1095 appear as locally attached devices to client systems 1010, 1020, and 1030. As with storage devices 1060(1)-(N) and storage devices 1070(1)-(N), storage devices 1090(1)-(N) and intelligent storage array 1095 generally represent any type or form of storage device or medium capable of storing data and/or other computer-readable instructions.

In certain embodiments, and with reference to exemplary computing system 910 of FIG. 9, a communication interface, such as communication interface 922 in FIG. 9, may be used to provide connectivity between each client system 1010, 1020, and 1030 and network 1050. Client systems 1010, 1020, and 1030 may be able to access information on server 1040 or 1045 using, for example, a web browser or other client software. Such software may allow client systems 1010, 1020, and 1030 to access data hosted by server 1040, server 1045, storage devices 1060(1)-(N), storage devices 1070(1)-(N), storage devices 1090(1)-(N), or intelligent storage array 1095. Although FIG. 10 depicts the use of a network (such as the Internet) for exchanging data, the embodiments described and/or illustrated herein are not limited to the Internet or any particular network-based environment.

In at least one embodiment, all or a portion of one or more of the exemplary embodiments disclosed herein may be encoded as a computer program and loaded onto and executed by server 1040, server 1045, storage devices 1060(1)-(N), storage devices 1070(1)-(N), storage devices 1090(1)-(N), intelligent storage array 1095, or any combination thereof.

All or a portion of one or more of the exemplary embodiments disclosed herein may also be encoded as a computer program, stored in server 1040, run by server 1045, and distributed to client systems 1010, 1020, and 1030 over network 1050.
As detailed above, computing system 910 and/or one or more components of network architecture 1000 may perform and/or be a means for performing, either alone or in combination with other elements, one or more steps of an exemplary method for aggregating information-asset metadata from multiple disparate data-management systems.

While the foregoing disclosure sets forth various embodiments using specific block diagrams, flowcharts, and examples, each block diagram component, flowchart step, operation, and/or component described and/or illustrated herein may be implemented, individually and/or collectively, using a wide range of hardware, software, or firmware (or any combination thereof) configurations. In addition, any disclosure of components contained within other components should be considered exemplary in nature since many other architectures can be implemented to achieve the same functionality.

In some examples, all or a portion of exemplary system 100 in FIG. 1 may represent portions of a cloud-computing or network-based environment. Cloud-computing environments may provide various services and applications via the Internet. These cloud-based services (e.g., software as a service, platform as a service, infrastructure as a service, etc.) may be accessible through a web browser or other remote interface. Various functions described herein may be provided through a remote desktop environment or any other cloud-based computing environment.

In various embodiments, all or a portion of exemplary system 100 in FIG. 1 may facilitate multi-tenancy within a cloud-based computing environment. In other words, the software modules described herein may configure a computing system (e.g., a server) to facilitate multi-tenancy for one or more of the functions described herein. For example, one or more of the software modules described herein may program a server to enable two or more clients (e.g., customers) to share an application that is running on the server. A server programmed in this manner may share an application, operating system, processing system, and/or storage system among multiple customers (i.e., tenants). One or more of the modules described herein may also partition data and/or configuration information of a multi-tenant application for each customer such that one customer cannot access data and/or configuration information of another customer.

According to various embodiments, all or a portion of exemplary system 100 in FIG. 1 may be implemented within a virtual environment. For example, the modules and/or data described herein may reside and/or execute within a virtual machine. As used herein, the
term "virtual machine" generally refers to any operating system environment that is abstracted from computing hardware by a virtual machine manager (e.g., a hypervisor). Additionally or alternatively, the modules and/or data described herein may reside and/or execute within a virtualization layer. As used herein, the term "virtualization layer" generally refers to any data layer and/or application layer that overlays and/or is abstracted from an operating system environment. A virtualization layer may be managed by a software virtualization solution (e.g., a file system filter) that presents the virtualization layer as though it were part of an underlying base operating system. For example, a software virtualization solution may redirect calls that are initially directed to locations within a base file system and/or registry to locations within a virtualization layer.

In some examples, all or a portion of exemplary system 100 in FIG. 1 may represent portions of a mobile computing environment. Mobile computing environments may be implemented by a wide range of mobile computing devices, including mobile phones, tablet computers, e-book readers, personal digital assistants, wearable computing devices (e.g., computing devices with a head-mounted display, smartwatches, etc.), and the like. In some examples, mobile computing environments may have one or more distinct features, including, for example, reliance on battery power, presenting only one foreground application at any given time, remote management features, touchscreen features, location and movement data (e.g., provided by Global Positioning Systems, gyroscopes, accelerometers, etc.), restricted platforms that restrict modifications to system-level configurations and/or that limit the ability of third-party software to inspect the behavior of other applications, controls to restrict the installation of applications (e.g., to only originate from approved application stores), etc. Various functions described herein may be provided for a mobile computing environment and/or may interact with a mobile computing environment.

In addition, all or a portion of exemplary system 100 in FIG. 1 may represent portions of, interact with, consume data produced by, and/or produce data consumed by one or more systems for information management. As used herein, the term "information management" may refer to the protection, organization, and/or storage of data. Examples of systems for information management may include, without limitation, storage systems, backup systems, archival systems, replication systems, high availability systems, data search systems, virtualization systems, and the like.
In some embodiments, all or a portion of exemplary system 100 in FIG. 1 may represent portions of, produce data protected by, and/or communicate with one or more systems for information security. As used herein, the term “information security” may refer to the control of access to protected data. Examples of systems for information security may include, without limitation, systems providing managed security services, data loss prevention systems, identity authentication systems, access control systems, encryption systems, policy compliance systems, intrusion detection and prevention systems, electronic discovery systems, and the like.

According to some examples, all or a portion of exemplary system 100 in FIG. 1 may represent portions of, communicate with, and/or receive protection from one or more systems for endpoint security. As used herein, the term "endpoint security" may refer to the protection of endpoint systems from unauthorized and/or illegitimate use, access, and/or control. Examples of systems for endpoint protection may include, without limitation, anti-malware systems, user authentication systems, encryption systems, privacy systems, spam-filtering services, and the like.

The process parameters and sequence of steps described and/or illustrated herein are given by way of example only and can be varied as desired. For example, while the steps illustrated and/or described herein may be shown or discussed in a particular order, these steps do not necessarily need to be performed in the order illustrated or discussed. The various exemplary methods described and/or illustrated herein may also omit one or more of the steps described or illustrated herein or include additional steps in addition to those disclosed.

While various embodiments have been described and/or illustrated herein in the context of fully functional computing systems, one or more of these exemplary embodiments may be distributed as a program product in a variety of forms, regardless of the particular type of computer-readable media used to actually carry out the distribution. The embodiments disclosed herein may also be implemented using software modules that perform certain tasks. These software modules may include script, batch, or other executable files that may be stored on a computer-readable storage medium or in a computing system.

In some embodiments, these software modules may configure a computing system to perform one or more of the exemplary embodiments disclosed herein.
In addition, one or more of the modules described herein may transform data, physical devices, and/or representations of physical devices from one form to another. For example, one or more of the modules recited herein may receive metadata of information assets from multiple disparate data-management systems that manage the information assets, transform the metadata into a global metadata repository, output a result of the transformation to a system that is capable of managing, utilizing, and enriching the global metadata repository, use the result of the transformation to provide a unified view of the information assets to the data-management systems, and store the result of the transformation to a storage system. Additionally or alternatively, one or more of the modules recited herein may transform a processor, volatile memory, non-volatile memory, and/or any other portion of a physical computing device from one form to another by executing on the computing device, storing data on the computing device, and/or otherwise interacting with the computing device.

The preceding description has been provided to enable others skilled in the art to best utilize various aspects of the exemplary embodiments disclosed herein. This exemplary description is not intended to be exhaustive or to be limited to any precise form disclosed. Many modifications and variations are possible without departing from the spirit and scope of the instant disclosure. The embodiments disclosed herein should be considered in all respects illustrative and not restrictive. Reference should be made to the appended claims and their equivalents in determining the scope of the instant disclosure.

Unless otherwise noted, the terms "connected to" and "coupled to" (and their derivatives), as used in the specification and claims, are to be construed as permitting both direct and indirect (i.e., via other elements or components) connection. In addition, the terms "a" or "an," as used in the specification and claims, are to be construed as meaning "at least one of." Finally, for ease of use, the terms "including" and "having" (and their derivatives), as used in the specification and claims, are interchangeable with and have the same meaning as the word "comprising."
WHAT IS CLAIMED IS:

1. A computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems, at least a portion of the method being performed by a computing device comprising at least one processor, the method comprising:
   - receiving, from a first data-management system that manages information assets of an entity in a first domain, a first instance of metadata of an information asset, the first instance of metadata having been generated by the first data-management system;
   - receiving, from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain, a second instance of metadata of the information asset, the second instance of metadata having been generated by the second data-management system;
   - storing the first instance of metadata and the second instance of metadata in a global metadata repository that is separate and distinct from the first data-management system and the second data-management system such that the first instance of metadata and the second instance of metadata are associated with the information asset;
   - providing access to the first instance of metadata and the second instance of metadata stored in the global metadata repository to one or more of:
     - the first data-management system;
     - the second data-management system;
     - the entity.

2. The method of claim 1, wherein:
   - the first data-management system manages a first instance of the information asset;
   - the second data-management system manages a second instance of the information asset;
   - the first instance of metadata comprises metadata of the first instance of the information asset;
   - the second instance of metadata comprises metadata of the second instance of the information asset;
storing the first instance of metadata and the second instance of metadata in the global metadata repository comprises:

determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset;

associating, based at least in part on determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset, the first instance of metadata and the second instance of metadata.

3. The method of claim 1, wherein:

the first data-management system manages a first instance of the information asset;

the second data-management system manages a second instance of the information asset;

the first instance of metadata comprises metadata of the first instance of the information asset;

the second instance of metadata comprises metadata of the second instance of the information asset;

storing the first instance of metadata and the second instance of metadata in the global metadata repository comprises:

generating a first fingerprint based at least in part on at least one of:

the first instance of the information asset;

the first instance of metadata;

generating a second fingerprint based at least in part on at least one of:

the second instance of the information asset;

the second instance of metadata;

determining that the first fingerprint matches the second fingerprint;

associating, based at least in part on determining that the first fingerprint matches the second fingerprint, the first instance of metadata and the second instance of metadata.
4. The method of claim 1, wherein the global metadata repository uses a normalized schema to store the first instance of metadata and the second instance of metadata.

5. The method of claim 1, wherein:
   - the global metadata repository uses an extensible schema to store the first instance of metadata and the second instance of metadata;
   - the extensible schema enables the first data-management system and the second data-management system to store any metadata in the global metadata repository.

6. The method of claim 1, wherein:
   - the information asset is accessible via an information-asset source;
   - the first data-management system and the second data-management system can generate the first instance of metadata;
   - the information-asset source must be accessed to generate the first instance of metadata;
   - the method further comprises coordinating the first data-management system and the second data-management system so that:
     - the first data-management system:
       - accesses the information-asset source;
       - generates the first instance of metadata;
       - stores the first instance of metadata in the global metadata repository;
     - the second data-management system accesses the first instance of metadata from the global metadata repository without accessing the information-asset source.

7. The method of claim 1, wherein:
   - the information asset is one of a plurality of information assets of the entity that are accessible via an information-asset source;
   - the steps of receiving the first instance of metadata and receiving the second instance of metadata comprise ensuring that metadata of each of the plurality of information assets is received from at least one of the first data-management system and the second data-management system.
8. The method of claim 1, wherein:
   the information asset is one of a plurality of information assets of the entity that are accessible via an information-asset source;
   the first data-management system comprises a backup system that has generated a backup of the information-asset source;
   receiving the first instance of metadata comprises requesting, from the backup system, an inventory of each information asset that is contained in the backup of the information-asset source.

9. The method of claim 1, wherein the information asset comprises at least one of:
   a file;
   an email;
   a document.

10. A computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems, at least a portion of the method being performed by a computing device comprising at least one processor, the method comprising:
   identifying, at a first data-management system, an information asset, wherein the first data-management system manages information assets of an entity in a first domain;
   generating, at the first data-management system, a first instance of metadata of the information asset;
   storing the first instance of metadata in a global metadata repository that:
       stores a second instance of metadata of the information asset that was collected from a second data-management system that:
           generated the second instance of metadata; and
           manages the information assets of the entity in a second domain that
           is separate and distinct from the first domain;
           is separate and distinct from the first data-management system and the second data-management system.
11. The method of claim 10, wherein:
the information asset is accessible via an information-asset source;
the first data-management system comprises a backup system that backs up the information-asset source;
the backup system generates the first instance of metadata as part of backing up the information-asset source.

12. The method of claim 10, further comprising:
detecting that the information asset has been modified;
updating, in response to detecting that the information asset has been modified, the first instance of metadata;
 storing the updated first instance of metadata to the global metadata repository.

13. The method of claim 10, further comprising:
requesting, at the first data-management system, information associated with the information asset from the global metadata repository;
receiving, at the first data-management system and from the global metadata repository, information associated with the information asset that includes the second instance of metadata;
performing, at the first data-management system, an action involving the information asset based at least in part on the information associated with the information asset received from the global metadata repository.

14. A system for aggregating information-asset metadata from multiple disparate data-management systems, the system comprising:
a receiving module, stored in memory, that:
receives, from a first data-management system that manages information assets of an entity in a first domain, a first instance of metadata of an information asset, the first instance of metadata having been generated by the first data-management system;
receives, from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain, a second instance of metadata of the information asset, the second instance of metadata having been generated by the second data-management system;

a storing module, stored in memory, that stores the first instance of metadata and the second instance of metadata in a global metadata repository that is separate and distinct from the first data-management system and the second data-management system such that the first instance of metadata and the second instance of metadata are associated with the information asset;

a providing module, stored in memory, that provides access to the first instance of metadata and the second instance of metadata stored in the global metadata repository to one or more of:

- the first data-management system;
- the second data-management system;
- the entity;

at least one processor that executes the receiving module, the storing module, and the providing module.

15. The system of claim 14, wherein:

the first data-management system manages a first instance of the information asset;

the second data-management system manages a second instance of the information asset;

the first instance of metadata comprises metadata of the first instance of the information asset;

the second instance of metadata comprises metadata of the second instance of the information asset;

the storing module stores the first instance of metadata and the second instance of metadata in the global metadata repository by:

determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset;
associating, based at least in part on determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset, the first instance of metadata and the second instance of metadata.

16. The system of claim 14, wherein:
the first data-management system manages a first instance of the information asset;
the second data-management system manages a second instance of the information asset;
the first instance of metadata comprises metadata of the first instance of the information asset;
the second instance of metadata comprises metadata of the second instance of the information asset;
the storing module stores the first instance of metadata and the second instance of metadata in the global metadata repository by:
generating a first fingerprint based at least in part on at least one of:
the first instance of the information asset;
the first instance of metadata;
generating a second fingerprint based at least in part on at least one of:
the second instance of the information asset;
the second instance of metadata;
determining that the first fingerprint matches the second fingerprint;
associating, based at least in part on determining that the first fingerprint matches the second fingerprint, the first instance of metadata and the second instance of metadata.

17. The system of claim 14, wherein the global metadata repository uses a normalized schema to store the first instance of metadata and the second instance of metadata.
18. The system of claim 14, wherein:
   the global metadata repository uses an extensible schema to store the first instance
   of metadata and the second instance of metadata;
   the extensible schema enables the first data-management system and the second
data-management system to store any metadata in the global metadata repository.

19. The system of claim 14, wherein:
   the information asset is accessible via an information-asset source;
   the first data-management system and the second data-management system can
   generate the first instance of metadata;
   the information-asset source must be accessed to generate the first instance of
   metadata;
   the receiving module and the providing module further coordinate the first data-
   management system and the second data-management system so that:
   the first data-management system:
      accesses the information-asset source;
      generates the first instance of metadata;
      stores the first instance of metadata in the global metadata repository;
   the second data-management system accesses the first instance of metadata
   from the global metadata repository without accessing the information-asset source.

20. The system of claim 14, wherein:
   the information asset is one of a plurality of information assets of the entity that are
   accessible via an information-asset source;
   the receiving module ensures that metadata of each of the plurality of information
   assets is received from at least one of the first data-management system and the second data-
   management system as part of receiving the first instance of metadata and receiving the
   second instance of metadata.
1. (Currently Amended) A computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems, at least a portion of the method being performed by a computing device comprising at least one processor, the method comprising:

   receiving, via an application programming interface that enables a first data-management system that manages information assets of an entity in a first domain and a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain to push information-asset metadata to and pull information-asset metadata from a global metadata repository that is separate and distinct from the first data-management system and the second data-management system, a first instance of metadata of an information asset from the first data-management system, the first instance of metadata having been generated by the first data-management system;

   receiving, via the application programming interface, a second instance of metadata of the information asset from the second data-management system, the second instance of metadata having been generated by the second data-management system;

   storing the first instance of metadata and the second instance of metadata in the global metadata repository such that the first instance of metadata and the second instance of metadata are associated with the information asset;

   providing, via the application programming interface, access to the first instance of metadata and the second instance of metadata stored in the global metadata repository to one or more of:

   the first data-management system;
the second data-management system;
the entity.

2. (Original) The method of claim 1, wherein:
the first data-management system manages a first instance of the information asset;
the second data-management system manages a second instance of the information asset;
the first instance of metadata comprises metadata of the first instance of the information asset;
the second instance of metadata comprises metadata of the second instance of the information asset;
storing the first instance of metadata and the second instance of metadata in the global metadata repository comprises:

determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset;
associating, based at least in part on determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset, the first instance of metadata and the second instance of metadata.

3. (Original) The method of claim 1, wherein:
the first data-management system manages a first instance of the information asset;
the second data-management system manages a second instance of the information asset;
the first instance of metadata comprises metadata of the first instance of the information asset;
the second instance of metadata comprises metadata of the second instance of the information asset;

storing the first instance of metadata and the second instance of metadata in the global metadata repository comprises:

- generating a first fingerprint based at least in part on at least one of:
  - the first instance of the information asset;
  - the first instance of metadata;
- generating a second fingerprint based at least in part on at least one of:
  - the second instance of the information asset;
  - the second instance of metadata;
- determining that the first fingerprint matches the second fingerprint;
- associating, based at least in part on determining that the first fingerprint matches the second fingerprint, the first instance of metadata and the second instance of metadata.

4. (Original) The method of claim 1, wherein the global metadata repository uses a normalized schema to store the first instance of metadata and the second instance of metadata.

5. (Original) The method of claim 1, wherein:

the global metadata repository uses an extensible schema to store the first instance of metadata and the second instance of metadata;
the extensible schema enables the first data-management system and the second data-management system to store any metadata in the global metadata repository.

6. (Original) The method of claim 1, wherein:

the information asset is accessible via an information-asset source;

the first data-management system and the second data-management system can generate the first instance of metadata;

the information-asset source must be accessed to generate the first instance of metadata;

the method further comprises coordinating the first data-management system and the second data-management system so that:

the first data-management system:

   accesses the information-asset source;

   generates the first instance of metadata;

   stores the first instance of metadata in the global metadata repository;

the second data-management system accesses the first instance of metadata from the global metadata repository without accessing the information-asset source.

7. (Original) The method of claim 1, wherein:

the information asset is one of a plurality of information assets of the entity that are accessible via an information-asset source;

the steps of receiving the first instance of metadata and receiving the second instance of metadata comprise ensuring that metadata of each of the plurality of information assets is
received from at least one of the first data-management system and the second data-
management system.

8. (Original) The method of claim 1, wherein:

the information asset is one of a plurality of information assets of the entity that are acces-
sible via an information-asset source;

the first data-management system comprises a backup system that has generated a backup of the information-asset source;

receiving the first instance of metadata comprises requesting, from the backup system, an inventory of each information asset that is contained in the backup of the information-asset source.

9. (Original) The method of claim 1, wherein the information asset comprises at least one of:

a file;
an email;
a document.

10. (Currently Amended) A computer-implemented method for aggregating information-asset metadata from multiple disparate data-management systems, at least a portion of the method being performed by a computing device comprising at least one processor, the method comprising:

identifying, at a first data-management system, an information asset, wherein the first data-management system manages information assets of an entity in a first domain;
generating, at the first data-management system, a first instance of metadata of the information asset;

storing, by the first data-management system via an application programming interface that enables the first data-management system and a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain to push information-asset metadata to and pull information-asset metadata from a global metadata repository that is separate and distinct from the first data-management system and the second data-management system and that stores a second instance of metadata of the information asset that was collected from the second data-management system that generated the second instance of metadata, the first instance of metadata in the global metadata repository.

11. (Original) The method of claim 10, wherein:

the information asset is accessible via an information-asset source;

the first data-management system comprises a backup system that backs up the information-asset source;

the backup system generates the first instance of metadata as part of backing up the information-asset source.

12. (Currently Amended) The method of claim 10, further comprising:

detecting that the information asset has been modified;

updating, in response to detecting that the information asset has been modified, the first instance of metadata;
storing, via the application programming interface, the updated first instance of
metadata to the global metadata repository.

13.  (Currently Amended) The method of claim 10, further comprising:

requesting, at the first data-management system via the application programming
interface, information associated with the information asset from the global metadata
repository;

receiving, at the first data-management system and from the global metadata
repository via the application programming interface, information associated with the
information asset that includes the second instance of metadata;

performing, at the first data-management system, an action involving the information
asset based at least in part on the information associated with the information asset received
from the global metadata repository.

14.  (Currently Amended) A system for aggregating information-asset metadata
from multiple disparate data-management systems, the system comprising:

a receiving module, stored in memory, that:

receives, via an application programming interface that enables a first data-
management system that manages information assets of an entity in a first domain
and a second data-management system that manages the information assets of the
entity in a second domain that is separate and distinct from the first domain to push
information-asset metadata to and pull information-asset metadata from a global
metadata repository that is separate and distinct from the first data-management
system and the second data-management system, a first instance of metadata of an
information asset from the first data-management system, the first instance of metadata having been generated by the first data-management system;

receives, via the application programming interface, a second instance of metadata of the information asset from the second data-management system, the second instance of metadata having been generated by the second data-management system;

a storing module, stored in memory, that stores the first instance of metadata and the second instance of metadata in the global metadata repository such that the first instance of metadata and the second instance of metadata are associated with the information asset;

a providing module, stored in memory, that provides, via the application programming interface, access to the first instance of metadata and the second instance of metadata stored in the global metadata repository to one or more of:

the first data-management system;

the second data-management system;

the entity;

at least one processor that executes the receiving module, the storing module, and the providing module.

15. (Original) The system of claim 14, wherein:

the first data-management system manages a first instance of the information asset;

the second data-management system manages a second instance of the information asset;

the first instance of metadata comprises metadata of the first instance of the information asset;
the second instance of metadata comprises metadata of the second instance of the information asset;

the storing module stores the first instance of metadata and the second instance of metadata in the global metadata repository by:

- determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset;
- associating, based at least in part on determining that the first instance of the information asset and the second instance of the information asset are instances of the same information asset, the first instance of metadata and the second instance of metadata.

16. (Original) The system of claim 14, wherein:

the first data-management system manages a first instance of the information asset;

the second data-management system manages a second instance of the information asset;

the first instance of metadata comprises metadata of the first instance of the information asset;

the second instance of metadata comprises metadata of the second instance of the information asset;

the storing module stores the first instance of metadata and the second instance of metadata in the global metadata repository by:

- generating a first fingerprint based at least in part on at least one of:
  - the first instance of the information asset;
  - the first instance of metadata;
generating a second fingerprint based at least in part on at least one of:

- the second instance of the information asset;
- the second instance of metadata;

determining that the first fingerprint matches the second fingerprint;

associating, based at least in part on determining that the first fingerprint matches the second fingerprint, the first instance of metadata and the second instance of metadata.

17. (Original) The system of claim 14, wherein the global metadata repository uses a normalized schema to store the first instance of metadata and the second instance of metadata.

18. (Original) The system of claim 14, wherein:

- the global metadata repository uses an extensible schema to store the first instance of metadata and the second instance of metadata;
- the extensible schema enables the first data-management system and the second data-management system to store any metadata in the global metadata repository.

19. (Original) The system of claim 14, wherein:

- the information asset is accessible via an information-asset source;
- the first data-management system and the second data-management system can generate the first instance of metadata;
- the information-asset source must be accessed to generate the first instance of metadata;

AMENDED SHEET (ARTICLE 19)
the receiving module and the providing module further coordinate the first data-
management system and the second data-management system so that:

the first data-management system:

accesses the information-asset source;

generates the first instance of metadata;

stores the first instance of metadata in the global metadata repository;

the second data-management system accesses the first instance of metadata
from the global metadata repository without accessing the information-asset source.

20. (Original) The system of claim 14, wherein:

the information asset is one of a plurality of information assets of the entity that are accessible via an information-asset source;

the receiving module ensures that metadata of each of the plurality of information assets is received from at least one of the first data-management system and the second data-
management system as part of receiving the first instance of metadata and receiving the second instance of metadata.
FIG. 1

System 100
- Modules 102
  - Receiving Module 104
  - Storing Module 106
  - Providing Module 108
  - Identifying Module 110
  - Generating Module 112
- Global Metadata Repository 120
  - Information Assets 122
  - Metadata 124
  - Fingerprints 126
  - Data-Management Systems 128
  - Information-Asset Sources 130

Information Asset 140
Start

Receive a first instance of metadata of an information asset from a first data-management system that manages information assets of an entity in a first domain

Receive a second instance of metadata of the information asset from a second data-management system that manages the information assets of the entity in a second domain that is separate and distinct from the first domain

Store the first and second instances of metadata in a global metadata repository that is separate and distinct from the first and second data-management systems

Provide access to the first and second instances of metadata stored in the global metadata repository to the first data-management system, the second data-management system, and/or the entity

End

FIG. 3
FIG. 6
FIG. 7
Start

Identify, at a data-management system, an information asset

Generate, at the data-management system, a first instance of metadata of the information asset

Store the first instance of metadata in a global metadata repository

End

FIG. 8
### A. CLASSIFICATION OF SUBJECT MATTER

INV. G06Q10/00

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

### B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

- **X** later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- **Y** document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- **Z** document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

Date of the actual completion of the international search: 13 July 2015

Date of mailing of the international search report: 22/07/2015

Name and mailing address of the ISA/Authorized officer:

European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk
Tel: (+31-70) 340-2040, Fax: (+31-70) 340-3016

Kirchler, Dominik
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2014006244 A1</td>
<td>02-01-2014</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>US 2014081890 A1</td>
<td>20-03-2014</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>